

---

# LPI exam prep: Mail and news

## Intermediate Level Administration (LPIC-2) topic 206

Skill Level: Intermediate

[David Mertz, Ph.D. \(mertz@gnosis.cx\)](mailto:mertz@gnosis.cx)

Developer

Gnosis Software

22 Nov 2005

This is the second of seven tutorials covering intermediate network administration on Linux®. In this tutorial, David Mertz discusses how to use Linux as a mail server and as a news server. Overall, e-mail is probably the main use of the Internet, and Linux is perhaps the best platform for e-mail services. This tutorial covers mail transport, local mail filtering, and mailing list maintenance software. It also briefly discusses server software for the NNTP protocol.

## Section 1. Before you start

Learn what these tutorials can teach you and how you can get the most from them.

### About this series

The [Linux Professional Institute](#) (LPI) certifies Linux system administrators at two levels: *junior level* (also called "certification level 1") and *intermediate level* (also called "certification level 2"). To attain certification level 1, you must pass exams 101 and 102; to attain certification level 2, you must pass exams 201 and 202.

developerWorks offers tutorials to help you prepare for each of the four exams. Each exam covers several topics, and each topic has a corresponding self-study tutorial on developerWorks. For LPI exam 202, the seven topics and corresponding developerWorks tutorials are:

Table 1. LPI exam 202: Tutorials and topics		
LPI exam 202 topic	developerWorks tutorial	Tutorial summary

Topic 205	<a href="#">LPI exam 202 prep (topic 205): Networking configuration</a>	Learn how to configure a basic TCP/IP network, from the hardware layer (usually Ethernet, modem, ISDN, or 802.11) through the routing of network addresses.
Topic 206	LPI exam 202 prep (topic 206): Mail and news	(This tutorial) Learn how to use Linux as a mail server and as a news server. Learn about mail transport, local mail filtering, mailing list maintenance software, and server software for the NNTP protocol. See detailed <a href="#">objectives</a> below.
Topic 207	LPI exam 202 prep (topic 207): DNS	Coming soon
Topic 208	LPI exam 202 prep (topic 208): Web services	Coming soon
Topic 210	LPI exam 202 prep (topic 210): Network client management	Coming soon
Topic 212	LPI exam 202 prep (topic 212): System security	Coming soon
Topic 214	LPI exam 202 prep (topic 214): Network troubleshooting	Coming soon

To start preparing for certification level 1, see the [developerWorks tutorials for LPI exam 101](#). To prepare for the other exam in certification level 2, see the [developerWorks tutorials for LPI exam 201](#). Read more about the [entire set of developerWorks LPI tutorials](#).

The Linux Professional Institute does not endorse any third-party exam preparation material or techniques in particular. For details, please contact [info@lpi.org](mailto:info@lpi.org).

## About this tutorial

Welcome to "Mail and news," the second of seven tutorials covering intermediate network administration on Linux. In this tutorial, you learn how to use Linux as a mail server and as a news server. This tutorial covers mail transport, local mail filtering, and mailing list maintenance software. It also briefly discusses server software for the NNTP protocol.

This tutorial is organized according to the LPI objectives for this topic. Very roughly, expect more questions on the exam for objectives with higher weight.

Table 2. Mail and news: Exam objectives covered in this tutorial		
LPI exam objective	Objective weight	Objective summary
2.206.1 <a href="#">Configuring mailing lists</a>	Weight 1	Install and maintain mailing lists using Majordomo. Monitor Majordomo problems by viewing Majordomo logs.
2.206.2 <a href="#">Using Sendmail</a>	Weight 4	Manage a Sendmail configuration including e-mail aliases, mail quotas, and virtual mail domains. This objective includes configuring internal mail relays and monitoring SMTP servers.
2.206.3 <a href="#">Managing mail traffic</a>	Weight 3	Implement client mail management software to filter, sort, and monitor incoming user mail. This objective includes using software such as Procmail on both the server and client side.
2.206.4 <a href="#">Serving news</a>	Weight 1	Install and configure news servers using INN. This objective includes customizing and monitoring served newsgroups.

## Prerequisites

To get the most from this tutorial, you should already have a basic knowledge of Linux and a working Linux system on which you can practice the commands covered in this tutorial.

---

## Section 2. About mail and news

The broad use of Linux for mail and news servers has led to the development of improved tools over time. When the LPI certification exams were developed, the most popular tools were *Sendmail* for mail transport, *Procmail* for local mail handling, *Majordomo* for mailing lists, and *innd* (InterNetNews daemon) for NNTP. The last of these is still probably the default choice for news; however, despite its technical strengths, the NNTP protocol has been somewhat eclipsed by e-mail mailing lists and Web-based discussion forums.

Of the other tools, Sendmail and Procmail are still widely used, although not as ubiquitously as they once were. The most popular upgrade or replacement for Sendmail is postfix, which contains facilities for backwards compatibility with

Sendmail. The local mail handling field is well populated with choices, but Procmail is still popular. On the other hand, Majordomo is a slight anachronism nowadays. Just as Majordomo largely replaced the earlier listserv software, mailman has more recently eclipsed Majordomo. However, to match the current LPI topic areas, this tutorial discusses Majordomo.

## Other resources

As with most Linux tools, it is always useful to examine the man pages for any utilities discussed. Versions and switches might change between utility or kernel version or with different Linux distributions. For more in-depth information, the Linux Documentation Project has a variety of useful documents, especially its HOWTOs. See the [Resources](#) section for a link. A variety of books on Linux networking have been published; I have found O'Reilly's *TCP/IP Network Administration*, by Craig Hunt, to be quite helpful (find whatever edition is most current when you read this; see [Resources](#) for a link).

---

## Section 3. Configuring mailing lists

### What does Majordomo do?

A mailing list manager program is basically a local extension for a mail transport program (MTA) such as Sendmail. Basically, the MTA running on a system passes off a set of addresses to the control of the mailing list manager, and the mailing list manager modifies, processes, and perhaps re-mails the messages it receives. Some messages received by a mailing list manager are messages meant for distribution to the mailing list itself (perhaps needing to be verified for permission to distribute to the list(s)). Other messages are control messages that change the status of the mailing list, such as the subscription options of a particular subscriber. A mailing list manager does not perform mail delivery itself, but passes that function to its supporting MTA.

As the introduction to this tutorial stated, Majordomo is not currently the state-of-the-art choice for mailing lists. Rather, the best choice for a new installation of a mailing list is probably Mailman. Majordomo, however, is still perfectly functional and is installed on many older systems which continue to operate without problem (sometimes supporting lists that have been operational for many years).

There is a wrinkle with Majordomo versions, however. Some years ago, a rewrite of the Majordomo 1.x series was started, called Majordomo2. Unfortunately, that rewrite fizzled out without ever reaching release status. While Majordomo2 (in a beta version) may be used in a very small number of systems, Majordomo 1.9.5 is the most recent stable version and is the version discussed in this tutorial.

## Installing Majordomo

You can obtain an archive of the Majordomo software at the Majordomo site (see [Resources](#) for a link).

After unpacking a file that will be named something like majordomo-1.94.5.tgz, be sure to read the INSTALL file carefully. You need to follow all the steps it describes for getting a working Majordomo system. Building the system uses the usual `make ; make install` steps of most source installs, as well as `make install-wrapper`. The install can and should verify itself with a command like `cd /usr/local/majordomo-1.94.5; ./wrapper config-test` (the `make install` provides details in a message).

Before building, modify the Makefile and create and/or modify majordomo.cf. As a starting point, you can copy the latter file from sample.cf in the source distribution. In the Makefile, a number of environment variables are set, but the most critical and subtle of these is probably `W_GROUP`. This is the *numeric* gid of the group Majordomo will run under, almost always the group "daemon." The gid for daemon is 1 on most systems, but be sure to check using the following:

```
$ id daemon
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

Other variables in Makefile include `PERL` for the path to the interpreter, and `W_HOME` for the location where Majordomo will be installed.

Your new majordomo.cf file also needs to be edited before the `make install`. The Perl variables that need to be modified appear mainly near the top of the file. Definitely adjust `$whereami` and `$homedir`, and examine the others to make sure they are sensible.

## Telling Sendmail to use Majordomo

The final step in installation is convincing Sendmail to talk with Majordomo. Within the `/etc/sendmail.cf` file, this involves a line like this:

```
OA/path/to/majordomo/majordomo.aliases
```

If you use the M4 processor to generate Sendmail configuration files, you can use a line like this:

```
define(`ALIAS_FILE',`/etc/aliases,/path/to/majordomo/majordomo.aliases')
```

The sample majordomo.aliases contains some sample values:

### Listing 1. Sample majordomo.aliases

```
majordomo:  "|/usr/test/majordomo-1.94.5/wrapper majordomo"
```

```
major-domo-owner: you
owner-major-domo: you
test:             "|/usr/test/major-domo-1.94.5/wrapper resend -l test test-list"
test-list:        :include:/usr/test/major-domo-1.94.5/lists/test
owner-test:       you
test-owner:       you
test-request:     you
```

These, of course, need to be customized for your particular setup. In particular, "you" means the name of the list administrator (who is not necessarily the overall system administrator).

## Creating a new Majordomo list

The sample setup given above created a list called "test," with addresses for "test-owner," "test-request," etc. for administering the list. In real use, you will probably want lists with other names. To accomplish that, do the following:

1. Switch to the directory `$listdir`, as defined in `major-domo.cf`.
2. Create files called `my-list-name` and `my-list-name.info` (adjust appropriately); `chmod` them to 664. The latter file contains an introduction to the list.
3. Create several aliases in your `major-domo.aliases` file, following the pattern of the "test" examples -- for example, "foo-owner," "foo," "foo-request," and so on.
4. Send requests to `subscribe`, `unsubscribe`, `signoff`, and so on, for members of the list.
5. Create an archive directory in the location specified by the `$filedir` and `$filedir_suffix` variables.
6. Create a digest subdirectory under `$digest_work_dir`. Use the same name as the digest list (for example: `test-digest`).
7. Make sure everything is owned by user `major-domo`, group `major-domo`, and is writeable by both owner and group (in other words, mode 664 for files and mode 775 for directories).
8. Issue a `config <listname> <listname>admin` command to Majordomo. This will cause it to create a default configuration file for the list, and send it back to you.

---

## Section 4. Using Sendmail

## What does Sendmail do?

Sendmail is a Mail Transport Agent (MTA). It routes, modifies, and delivers mail message across heterogeneous mail systems. With a history somewhat parallel to that of mailing list software, Sendmail has a "permanent beta" version called Sendmail X that is intended as an upgrade/replacement for the stable Sendmail 8.x series; however, much as Mailman has largely supplanted Majordomo, several MTAs have partially eclipsed Sendmail. The chief such new MTA is Postfix, but Qmail and Exim are also widely used. Nonetheless, Sendmail still remains, at least by a narrow margin, the most widely used MTA on Linux systems. As of September 16, 2005, the latest stable release of Sendmail was 8.13.5.

Not just one book, but many books, have been written on Sendmail. See [Resources](#) for a list of available books. The most comprehensive of these is *Sendmail, Third Edition* (O'Reilly, 2002) by Bryan Costales with Eric Allman. At 1,232 pages, this book covers quite a lot more than this tutorial can touch on.

While Sendmail in principle supports a number of mail transport protocols such as UUCP, by far the most widely used is Simple Mail Transport Protocol (SMTP), which here includes Extended SMTP (ESMTP) for enhanced MIME encoded message bodies. At heart, mail that is not forwarded to other SMTP hosts is delivered to the local system by putting messages in local files. Local Mail User Agents (MUAs) read messages that Sendmail (or another MTA) puts in local files (and often also fetch mail using POP3 or IMAP), but generally call on Sendmail to deliver outgoing messages. Some MUAs, however, directly communicate with SMTP servers (such as Sendmail instances, local or remote) rather than placing messages in the Sendmail queue for later processing. Usually the Sendmail queue is in `/var/spool/mqueue/`.

## Installing Sendmail

The first thing to do is obtain a copy of the current Sendmail software from [sendmail.org](http://sendmail.org) (see [Resources](#) for a link), for example, `sendmail.8.13.5.tar.gz`. Unpack it as usual. Unlike many applications that use the `make; make install` pattern, building Sendmail is performed with `sh Build`. After the initial build, `cd` to the `cf/cf/` subdirectory; copy a suitable `*.mc` file as `sendmail.mc`; customize `sendmail.mc`; and run the following to generate a `sendmail.cf` file:

```
$ m4 ../m4/cf.m4 sendmail.mc > sendmail.cf
```

You may also use the shortcut `sh Build sendmail.cf`. This may seem mysterious, but both these commands generate an actual Sendmail configuration from a more readable format using the M4 macro processor. Actual `sendmail.cf` files, though editable ASCII, are quite cryptic and should only be modified minimally by hand.

Finally, copy the sendmail binary from a location that will be something like

obj.Linux.2.6.10-5-386.i686/sendmail/sendmail to its final location (back up an old one if it exists), typically /usr/sbin/, and copy your sendmail.cf file to /etc/mail/sendmail.cf. You can also do the latter in the cf/cf/ subdirectory with `sh Build install-cf`. You will probably need to `su` or `sudo` to obtain file permissions for the relevant directories.

A number of utilities come with Sendmail: makemap, mailstats, etc. Each corresponding directory has a README and can be installed by running `sh Build install` from the subdirectory.

## The sendmail.cf file

The main complexity, and the main function, of Sendmail is in its sendmail.cf file. This configuration file contains some settings for the Sendmail environment, but principally it contains patterns for addresses to rewrite and/or deliver by certain mechanisms.

Two rewrite mechanism that may be configured are the `genericstable` and `virtusertable`, which let you map local users to and from external addresses. For either mapping, you first create an aliases file as plain text. For example:

### Listing 2. Outbound mappings

```
david      david.mertz@gmail.com
root      root@gnosis.cx
dqm@gnosis.lan  david.mertz@gmail.com
```

Or, for incoming mail mapped to local accounts:

### Listing 3. Inbound mappings

```
david@mail.gnosis.cx      david
david@smtp.gnosis.cx      david
david@otherdomain.net     david
@mail.gnosis.cx            %1@external-host.com
owner@list.gnosis.cx       owner%3
jax@bar.com                error:5.7.0:550 Address invalid
```

To compile these aliases, use the `makemap` utility:

```
$ makemap dbm /etc/mail/virtusertable < inbound
$ makemap hash /etc/mail/genericstable < outbound
```

Enabling use of these maps can be configured using M4 macros in sendmail.cf (or in whatever configuration file you use).

### Listing 4. Enabling mappings in sendmail.cf

```
DOMAIN(gnosis.cx)dnl
FEATURE(`virtusertable', `dbm /etc/mail/virtusertable')dnl
```



```
FEATURE('genericstable', `hash /etc/mail/genericstable')dnl
GENERIC_DOMAIN_FILE(`/etc/mail/generics-domains')dnl
```

A number of things are going on here. The `DOMAIN` macro indicates that a file like `cf/domain/gnosis.cx.m4` is used for additional macros. The `FEATURE` macros enable use of the `virtusertable` and `genericstable`. The `GENERIC_DOMAIN_FILE` macro defines the domains that qualify for remapping for names in `genericstable`.

Rewriting will follow all the rules indicated. In test mode (`sendmail -bt`), you can examine the rewriting that is performed for specific addresses. For example, using `genericstable`, mail to the local user "david" will be delivered to `david.mertz@gmail.com` externally. Assuming `localhost` is defined in `/etc/mail/generics-domains`, mail to `david@localhost` will go to the same place.

In the other direction, mail coming in for `david@mail.gnosis.cx` will be rewritten and delivered to local user "david." Multiple domains can be manipulated by Sendmail at the same time, so `david@otherdomain.net` will also be delivered locally.

The full power comes in some of the wildcard symbols. Any mail sent to `mail.gnosis.cx` that is not specifically directed to a local user will be forwarded to the same username at `external-host.com`. But that's a simple pattern. More interestingly, the `%3` can be used to expand multiple extra name information, so `owner-foo@list.gnosis.cx` and `owner-bar@list.gnosis.cx` will be delivered to local users "owner-foo" and "owner-bar," respectively (unless they undergo further rewriting). These local users might be mailing list processing systems or other automated message handlers. As a special case, you can raise an error for a given address rather than rewrite it further.

What we have looked at so far really just scratches the surface of the rewriting rules you can add to Sendmail, but they give you an initial feel. Buy one of the large books on the topic to learn more details.

## Running Sendmail

Sendmail can run in a number of modes. The most common mode is as a daemon that stays in the background and periodically process its queue. For example, running:

```
$ /usr/sbin/sendmail -bd -q10m
```

tells Sendmail to run as a daemon and check its queue every ten minutes. You can also run Sendmail a single time to process the queue at once, but not daemonize:

```
$ /usr/sbin/sendmail -q
```

As mentioned above, Sendmail has a test mode to examine address rewriting rules. For example (taken from the *Linux Network Administrators Guide*; see [Resources](#) for a link):

## Listing 5. Sendmail test mode

```
$ /usr/sbin/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 isaac@vstout.vbrew.com
rewrite: ruleset 3 input: isaac @ vstout . vbrew . com
rewrite: ruleset 96 input: isaac < @ vstout . vbrew . com >
rewrite: ruleset 96 returns: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 3 returns: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 0 input: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 199 input: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 199 returns: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 98 input: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 98 returns: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 198 input: isaac < @ vstout . vbrew . com . >
rewrite: ruleset 198 returns: $# local $: isaac
rewrite: ruleset 0 returns: $# local $: isaac
```

---

## Section 5. Managing e-mail traffic

### What does Procmail do?

Procmail is a mail processor. Basically, once Sendmail or another MTA has delivered mail to a local mailbox, you might use an MUA to process the mail in your inbox. You save some messages to various folders; you delete others; you forward other messages to various interested parties; you reply to others; and so on. Doing these tasks in an MUA is a manual and interactive process, and is potentially time consuming.

Procmail is a program that can do these tasks for you automatically whenever the required processing can be stated in a rule-driven way. Obviously, when you write back to your mother about her personal e-mail, some personal attention is required; but for a large class of other messages, it is useful to specify in advance exactly what you would like to happen when a given message is received. The rules that can drive automated message handling might involve specific pattern-based header fields, certain contents in a message body, or even calls out to more specific and specialized external programs such as statistical spam filters.

### Enabling Procmail

Procmail probably came installed with your Linux distribution. If not, you can obtain the source archive at [procmail.org](http://procmail.org) (see [Resources](#)). As of this writing, the latest version is 3.22. You may also be able to install Procmail as a binary using the install system of your Linux distribution (for example, on Debian: `apt-get install procmail`). Building from source is a straightforward `make install`. All Procmail needs to operate is the procmail binary and a `~/procmailrc` configuration file (or

possibly a global `/etc/procmailrc` file).

Beyond installing Procmail in the first place, you need to get your local mail system to utilize Procmail. An older mechanism to process mail through Procmail is to use a `.forward` file; this will still often work on a per-user basis. Usually, a user will create a file, `~/forward`, that contains something like this:

```
| /usr/local/bin/procmail
```

This will pipe each incoming message to Procmail. However, a better and more common way to utilize Procmail is to tell your MTA to talk directly to Procmail in the first place. In Sendmail, this involves enabling the `local_procmail` feature by putting the following in your `sendmail.mc` file:

```
FEATURE(`local_procmail', `'/usr/bin/procmail', `procmail -Y -a
$h -d $u')
```

Once Procmail is enabled, it needs a file, `~/procmailrc`, which contains the set of rules it processes in handling a given message. Procmail is not a daemon, but rather a text processing tool that accepts exactly one e-mail message at a time via STDIN.

## Rules in `~/procmailrc`

At heart, Procmail is just a set of regular expression recipes. You may also define environment variables in the same fashion as in a shell script. Recipes are executed in order, but flags may be used to execute a particular condition only if the prior condition is satisfied (`A`) or is not satisfied (`E`). Some Procmail recipes are delivery recipes, and others are non-delivery recipes; the former terminates processing of a given message, unless the `c` flag is given to explicitly continue processing. Probably the most common action of a recipe is to store a message in a named mailbox, but you may also pipe a message to another program or forward the message to a list of addresses.

A recipe usually starts with a lock (optionally with a specific lock file; otherwise, it is done automatically) and some flags, followed by some rules, and then by exactly one action. For example:

```
:0 [flags] [ : [locallockfile] ]
<zero or more conditions (one per line)>
<exactly one action line>
```

Of particular note are the implied flag `H` to match the header and `B` to match the body. Patterns normally are case-insensitive, but the `D` flag can force case-sensitive matching.

If a condition begins with `*`, everything after that character is an `egrep` expression. Otherwise, if a line starts with `<` or `>`, it checks the size of a message as being smaller or larger than a given number of bytes. The `$` prefix allows shell substitutions.

An action that is simply a file name saves a message to that mailbox. Use the special pseudo-file `/dev/null` to delete a message. A pipe character (`|`) passes the message to another program, such as the digest-splitting utility for mail that is distributed with Procmail. The exclamation prefix (`!`) forwards a message as an action (but negates a condition in a rule). Some examples:

### Listing 6. Sample `~/procmailrc` file

```
:0:
* ^Subject:.*Digest                # split digests and save parts
* ^From:.*foo-digest
|formail +1 -ds cat >>mailing_lists_mailbox

:0:
* !(To|Cc).*mertz@gnosis.cx        # my main account here
* !(To|Cc).*david.mertz@gmail.com  # I get mail from here
* !From.*gnosis\.*cx              # I trust gnosis not to spam
* !From.*list.*@                  # don't trash mailing lists
* !From.*good-buddy               # sometimes Bcc's me mail
spam

:0:
* ^Subject.*[MY-LIST]              # redistribute MY-LIST messages
! member@example.com, member2@example.net, member3@example.edu

:0:
* ^Cc.*joe@somewhere.org           # save to both inbox and JOE mbox
{
    :0 c
    $DEFAULT

    :0
    JOE
}
```

---

## Section 6. Serving NNTP news

### What does InterNetNews do?

NNTP is a nice protocol for "pull" distribution of messages to any users who are interested in a given topic. The Usenet is a large collection of "newsgroups" on thousands of different topics that distribute messages via NNTP. Being a pull protocol, an NNTP server gathers the current messages available from a decentralized network of servers, selecting only those newsgroups that the site administrator chooses to include. When a new message is posted to a given newsgroup, it propagates non-hierarchically from the server to all the other servers on the Internet interested in subscribing to that particular newsgroup.

From an end-user perspective, a mailing list can appear very similar to a newsgroup. In either case, the user composes and posts messages, and reads messages written by other people. In the ancient days of Usenet and the Internet, mailing lists were not as capable of presenting discussion topics in a "threaded" fashion as

newsgroups now do automatically. But for a number of years, mail clients have done a good job of inferring the discussion threads within mailing lists.

The main difference between newsgroups and mailing lists is in their underlying network protocol. A mailing list still relies on one centralized mail server that accepts all the messages destined for a particular list, and distributes that message via e-mail to all users who have indicated an interest (and have been approved, either by automatic or human-moderated subscription mechanisms). In contrast, NNTP connects every node to every other one without relying on a central server; each NNTP server simply talks to the other servers "nearby," and messages rather rapidly reach the whole world.

InterNetNews (INN) is an NNTP server that was first written in 1992, and has been actively maintained since then. As of this writing, INN is at version 2.4.1. The home page for INN includes releases and documentation (see [Resources](#) for a link).

## Setting up INN

After obtaining and unpacking the current source release, building INN is a straightforward `./configure; make; make install` sequence. To build INN, you need to have Perl and yacc (or bison) installed. This creates a number of files, mostly in the `/usr/local/news/` directory (which you probably do not have if INN has not been installed previously).

Before running the `innd` daemon (as user "news"), you should modify a number of configuration files. Full details are outside our scope, but a longer tutorial entitled *Installing and Running a Usenet News Server with INN and FreeBSD* on the full set of files needing attention is available online (see [Resources](#) for a link). Many of the permissions and quota issues are handled by the make system, but you might want to double check these configurations.

A file to pay particular attention to is the quota setup in `/usr/local/news/etc/storage.conf`. This controls which newsgroups are subscribed to and how much history from each newsgroup to maintain. Once the quota is reached, older messages are purged from a given newsgroup (on the local server, not from Usenet as a whole). For example, `storage.conf` might contain something like this:

### Listing 7. Sample `storage.conf` configuration

```
method cnfs {
  newsgroups: alt.binaries.*
  class: 1
  size: 0,1000000
  options: BINARIES
}

method cnfs {
  newsgroups: *
  class: 2
  size: 0,100000
  options: NOTBINRY
}
```

The `class` value specifies the order in which different rules are evaluated.

Once all the various configuration files are tweaked, just running `innd` as a daemon (probably launched for an initialization script) monitors the upstream servers configured by `/usr/local/news/etc/innfeed.conf`, `/usr/local/news/etc/incoming.conf`, and `/usr/local/news/etc/newsfeeds`.

# Resources

## Learn

- Review the entire [LPI exam prep tutorial series](#) on developerWorks to learn Linux fundamentals and prepare for system administrator certification.
- At the [LPIC Program](#), find task lists, sample questions, and detailed objectives for the three levels of the Linux Professional Institute's Linux system administration certification.
- View this [700 Linux User Groups around the world](#) -- many LUGs have local and distance study groups for LPI exams.
- The [Linux Documentation Project](#) has a variety of useful documents, especially its HOWTOs.
- *TCP/IP Network Administration, Third Edition* by Craig Hunt (O'Reilly, April 2002) is an excellent resource on Linux networking.
- The [Linux Network Administrators Guide](#) is an extensive online book covering many facets of Linux networking. Of specific interest for testing Sendmail is section 18.9, [Testing Your Configuration](#).
- The [Sendmail site](#) offers a [list of Sendmail books](#) for reference, including *Sendmail, Third Edition* (O'Reilly, December 2002).
- See [Installing and Running a Usenet News Server with INN and FreeBSD](#) for a more complete tutorial on installing and running an INN server.
- Find more [tutorials for Linux developers](#) in the [developerWorks Linux zone](#).

## Get products and technologies

- At the [Majordomo site](#), find downloads and information for Majordomo.
- At [procmail.org](#), download the latest version of Procmail.
- At [sendmail.org](#), download the latest version of Sendmail.
- Go to the [INN home page](#) for InterNetNews information and downloads.
- [Order the SEK for Linux](#), a two-DVD set containing the latest IBM trial software for Linux from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
- Build your next development project on Linux with [IBM trial software](#), available for download directly from developerWorks.

## Discuss

- [Participate in the discussion forum for this content](#).
- Browse the [developerWorks blogs](#) and get involved in the developerWorks community.

## About the author

David Mertz, Ph.D.

David Mertz has been writing the developerWorks columns *Charming Python* and *XML Matters* since 2000. Check out his book [Text Processing in Python](#) . For more on David, see his [personal Web page](#).